SmurfCat at SemEval-2025 Task 3: Bridging External Knowledge and Model Uncertainty for Enhanced Hallucination Detection

Elisei Rykov[1] **Valerii Olisov**[5] **Maksim Savkin**[5]
Artem Vazhentsev[1,2] **Kseniia Titova**[1,3] **Alexander Panchenko**[1,2]
Vasily Konovalov[2,5] **Julia Belikova**[4,5]
[1]Skoltech [2]AIRI [3]MTS AI [4]Sber AI Lab
[5]Moscow Institute of Physics and Technology
elisei.rykov@skol.tech belikova.iua@phystech.edu

Abstract

The Multilingual shared-task on Hallucinations and Related Observable Overgeneration Mistakes in the SemEval-2025 competition aims to detect hallucination spans in the outputs of instruction-tuned LLMs in a multilingual context. In this paper, we address the detection of span hallucinations by applying an ensemble of approaches. In particular, we synthesized a dataset and fine-tuned LLM to detect hallucination spans. In addition, we combined this approach with a white-box method based on uncertainty quantification techniques. Using our combined pipeline, we achieved 3rd place in detecting span hallucinations in Arabic, Catalan, Finnish, Italian, and ranked within the top ten for the rest of the languages.

1 Introduction

In recent years, there have been significant advancements in Natural Language Generation (NLG) models, mainly due to transformer-based architectures such as GPT (Radford et al., 2019). Nevertheless, the field faces two related challenges: the first is the propensity for current neural systems to create incorrect, yet coherent outputs, and the second is the inefficiency of current metrics in prioritizing accuracy over fluency. This leads to a phenomenon known as "hallucination", where NLG models generate cohere but inaccurate outputs that are difficult to automatically identify (Ji et al., 2023).

The shared-task on Multilingual Hallucinations and Related Observable Overgeneration Mistakes (Mu-SHROOM, Vázquez et al. (2025)) has been suggested to address this challenge. In particular, the Mu-SHROOM task aims to detect hallucination spans in the outputs of instruction-tuned LLMs in a multilingual context models (Arabic, Basque, Catalan, Chinese, Czech, English, Farsi, Finnish, French, German, Hindi, Italian, Spanish, and Swedish).[1]

[1]https://helsinki-nlp.github.io/shroom/

Mu-SHROOM is the continuation of the competitions in hallucination detection, the first being SHROOM. Its goal was to detect hallucinations and overgeneration errors within various generation tasks, such as machine translation, paraphrasing, and definition modeling (Maksimov et al., 2024; Rykov et al., 2024).

To address the Mu-SHROOM challenge, we developed an ensemble of approaches. First, we fine-tuned LLM on synthetic span-level hallucination detection data. Then, we combined this approach with white-box methods based on uncertainty quantification (UQ) techniques. Using our combined pipeline, we achieved 3rd place in detecting span hallucinations in Arabic, Catalan, Finnish, Italian, and ranked within the top ten for the rest of the languages.

Our contribution could be summarized as follows:

- We demonstrate a pipeline for synthetic data generation without any human annotation for span-level hallucination detection.
- We propose training an additional lightweight model that leverages multiple white-box UQ methods, demonstrating effectiveness without relying on any external information.
- We demonstrate that combining both the white-box and black-box methods can further enhance performance.

2 Related Work

2.1 Black-box

Within the scope of black-box approaches for hallucination detection, FactScore (Min et al., 2023) is a well-known approach. It first extracts atomic facts from the model's response and compares them to a retrieved context using an additional LLM. This process yields a fact-verification score that indi-

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1034–1045

cates whether the claims are supported by the retrieved context.

Several approaches exist for the detection of word-level hallucinations, among which RAGTruth (Niu et al., 2024) is a widely recognized pipeline for this task. Initially, the developed dataset and its corresponding benchmark were designed to evaluate LLMs within a retrieval-augmented generation (RAG) pipeline for various tasks, such as summarization, question-answering, and others. However, it could be easily adapted for the fact-checking task. The dataset was created using human annotation of LLM responses to capture hallucinations.

Furthermore, the task of hallucination detection can naturally be extended to hallucination editing. For example, the FAVA (Mishra et al., 2024) model is specifically trained for word-level hallucination detection and editing tasks according to the introduced hallucination taxonomy. To collect training data, the authors asked LLM to insert errors from the introduced taxonomy into the responses.

Despite their advantages, both RAGTruth and FAVA are limited in their applicability, as they are designed only for English-language tasks.

2.2 White-box

White-box approaches leverage internal generation signals, such as token-level probability distributions or hidden states, to detect hallucination of LLMs. For instance, Token Probability and Token Entropy (Fomicheva et al., 2020) utilize the probability distribution of each token. Belikova et al. (2024) demonstrated that token maximum probability and margin probability can be successfully used to enhance the trustworthiness of the RAG pipeline over knowledge bases. Moskvoretskii et al. (2025) showed that UQ scores can be used to develop adaptive RAG pipeline that outperforms the vanilla RAG in both performance and computational efficiency. Krayko et al. (2025) applied UQ scores, particularly mean token entropy and mean token probability, for the continuous evaluation of the RAG pipeline. Fadeeva et al. (2023) introduced the Claim Conditioned Probability (CCP) method, which evaluates the consistency of the several most probable token candidates.

These methods, while simple and effective for various tasks, exhibit several limitations in hallucination detection across multiple models. The distribution of UQ scores can differ between models. Furthermore, UQ scores are often poorly calibrated (Kadavath et al., 2022), which requires the use of an additional trainable model to normalize these values.

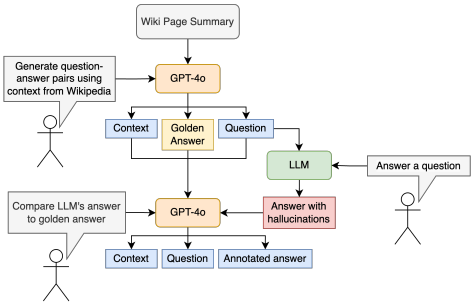

Figure 1: Synthetic data collection procedure. The detailed process is described in Section 3.

Studies have shown that hidden states (Azaria and Mitchell, 2023; CH-Wang et al., 2024; Vazhentsev et al., 2025) and attention matrices (Chuang et al., 2024; Vazhentsev et al., 2024) of LLMs contain significant information on the truthfulness of model output. These works suggest training auxiliary models to predict uncertainty using these features.

Various methods require multiple stochastic samples from LLMs to quantify uncertainty based on the consistency of generated answers (Manakul et al., 2023; Lin et al., 2023; Duan et al., 2024; Vashurin et al., 2025). Although these approaches are effective in sequence-level tasks, none of these methods can be directly applied to token-level hallucination detection.

3 Multilingual Synthetic Dataset for Hallucination Detection

All of our approaches require additional data for fine-tuning and calibration. The exact scheme of synthetic data generation is shown in Figure 1.

To collect synthetic data, we first generate multilingual question-context-answer triplets based on contexts from Wikipedia. Generation was performed using GPT-4o for the exact prompt used to generate question-answer pairs. Next, we collect the hypotheses by passing the generated questions without any contexts to various LLMs. Finally, we ask GPT-4o to find all inconsistencies between the LLM answer and the golden answer. Any inconsistent information in the hypotheses that contradicts the golden answer is considered a hallucination. In total, synthetic dataset contains 52 271 samples after all filtering stages.

The generation of synthetic training data through

LLM has been successfully employed to address detoxification (Moskovskiy et al., 2024, 2025) and PII detection (Savkin et al., 2025).

4 System Overview

4.1 Black-box

Our black-box approach incorporates a retriever that provides additional context along with the question-answer pair to a fine-tuned multilingual LLM. The LLM then highlights the spans in the answer that contain hallucinations. The architecture of the black-box pipeline is shown in Figure 2 in Appendix D.

A fine-tuned LLM, by itself, provides only hard labels, as it simply inserts special tokens around the spans with hallucinations. Therefore, to obtain soft labels with probabilities for each span, we employ two strategies. The first method is a `logit`-based approach, which assigns the probability of the span based on the probability of the opening tag. The next method is a `sampling`-based approach that samples the annotations from the model several times and aggregates the predictions. The probability of a span is calculated as the normalized frequency of its occurrence across samples. In addition, the baseline approach confidently set probability of 1.0 for each predicted span.

4.2 White-box

Our white-box pipeline is designed to predict token-level hallucination probabilities by leveraging uncertainty scores. For a given generated text $\tilde{y}$ of a length N, for each token $t_i \in \tilde{y}$, $i = 1 \dots N$, the procedure consists of the following steps:

1. We construct a feature vector $\mathbf{x}_i$ by concatenating the uncertainty scores obtained using a sliding window centered at the position i. For a given window size k, the feature vector is defined as:

$$\mathbf{x}_i = \bigoplus_{m=0}^{M-1} \left[\mathbf{u}_{i-k}^{(m)}, \mathbf{u}_{i-k+1}^{(m)}, \dots, \mathbf{u}_{i+k}^{(m)} \right],$$

   where $\bigoplus$ denotes the concatenation operation, $\mathbf{u}_j^{(m)}$ represents the uncertainty score from method m for token j, and M – total number of UQ methods. In our experiments, we use Token Probability, Token Entropy, and CCP. For tokens outside the valid range (i.e., if $j < 0$ or $j > N$), we set $\mathbf{u}_j^{(m)} = 0$.

2. The target label $y_i \in [0, 1]$ is defined as the maximum hallucination probability across all spans covering the token. If token i does not belong to any hallucinated span, we set $y_i = 0$.

Finally, we train a lightweight calibration model $f(\cdot)$ to predict the hallucination probability for each token, given its feature $\mathbf{x}_i$. For the model $f(\cdot)$, we employ logistic regression (LR) and gradient boosting (GB). To convert the soft probability predictions into binary hard labels, we determine an optimal probability threshold using the validation set.

4.3 Merging

To take advantage of both approaches, we integrate predictions from the white-box and black-box methods. For each token, the final hallucination probability is calculated as a weighted average:

$$p_{\text{final}} = \alpha\, p_{\text{black box}} + \beta\, p_{\text{white box}},$$

where $p_{\text{black box}}$ and $p_{\text{white box}}$ represent the probabilities obtained using the black-box and white-box methods, respectively, with α and β being positive tunable parameters that satisfy $\alpha + \beta = 1$.

5 Experimental Setup

5.1 Baselines

For all baselines, we adopt the context obtained in the retrieval stage for the black-box method, described in Section 5.2. Thus, we evaluated the FAVA[2] model passing questions and answers from the validation and test subsets along with the retrieved contexts. This ensures FAVA and black-box are tested on identical input data. We did not perform any soft labeling for FAVA, therefore, we assign a probability of 1.0 for each span.

Furthermore, we train the ModernBERT[3] encoder on the binary token classification task using our synthetic data. The training parameters are presented in Appendix E.

For FactScore, we use retrieved contexts to verify each generated atomic fact. For each token, the soft label equals the frequency of it appearing in unsupported claims or equals zero for all tokens that appear in all claims or are present in the original input.

[2] https://hf.co/fava-uw/fava-model
[3] https://hf.co/answerdotai/ModernBERT-large

We also select different LLMs as baselines, including both open-source and proprietary ones: GPT-4o, Phi-4[4], and Qwen2.5-7B-it. In the prompt, we asked models to identify and highlight hallucinations in the text using the tags `[HAL]` and `[/HAL]`. The evaluation is performed with retrieved context in 3-shot mode, where, for each question-answer pair, we randomly sample three examples of correctly identified hallucinations from the language-specific validation set.

5.2 Black-box

Model fine-tuning: As an LLM for the black-box hallucination detection pipeline, we fully fine-tuned Qwen2.5-7B-Instruct[5] due to its strong multilingual capabilities. See Appendix F for more details on LLM selection. For highlighting hallucination spans, we add two special tokens `[HAL]` and `[/HAL]` to the model's tokenizer. We added the synthetic data along with FAVA and RAGTruth to the training dataset mixture. In total, the model was trained with 84 334 training samples. Details on dataset mixture and training hyperparameters are presented in the Appendix B.

Retrieval: To retrieve the contexts for hallucination detection, we used the DuckDuckGo API[6]. We simply passed the question as is and collected the top 20 pages from the search output. Next, we filtered only Wikipedia articles related to the question from the search output. Finally, we collected and merged all Wikipedia summaries for the questions.

Soft Labeling: We compare different soft-labeling strategies:

- `Base`: simply assign a probability of 1.0 to each selected span.
- `Logit`: extract the probability of the opening `[HAL]` token in a greedy decoding setup.
- `Temp`: perform temperature sampling and set `top_k`, `top_p`, `num_beams`, and a `temperature` parameters.
- `DBS`: perform Diverse Beam Search (Vijayakumar et al., 2016) sampling strategy and adjust `diversity_penalty`, `num_beams`, and a `num_beam_groups` parameters.

5.3 White-box

We use the implementation of uncertainty quantification methods from LM-Polygraph (Fadeeva et al., 2023; Vashurin et al., 2024). In our experiments, we consider two training strategies for the calibration model.

Model-specific training: For each language, a separate hallucination detection model was trained using bootstrap-validation with a 70/30 ratio for train/validation split.

Model-agnostic training: Data from all languages were combined and a single model was trained using K-fold cross-validation. This approach yielded a more stable training process due to reduced data variance.

[4] https://hf.co/microsoft/phi-4
[5] https://hf.co/Qwen/Qwen2.5-7B-Instruct
[6] https://duckduckgo.com

Method	Mode	val		test	
		IoU	Cor	IoU	Cor
Black-box$_{\text{Base}}$	SFT	46.78	43.58	53.42	51.41
Black-box$_{\text{DBS}}$	SFT	53.43	49.95	56.56	**57.49**
White-box$_{\text{LR}}$	-	45.10	39.42	42.80	40.79
White-box$_{\text{GB}}$	-	48.29	44.95	45.69	43.67
Merging	-	**57.40**	**50.65**	**58.05**	52.88
FAVA	-	26.49	15.73	27.43	18.05
ModernBERT	SFT	32.87	30.13	33.35	32.55
FactScore$_{\text{GPT-4o}}$	-	22.52	16.65	24.05	20.69
GPT-4o	3-shot	-	-	49.07	46.77
Phi-4	3-shot	-	-	33.19	35.43
Qwen2.5-7B-it	3-shot	-	-	20.04	20.83

Table 1: Main results. For black-box, we report two soft-labeling strategies: `Base`, without any specific soft-labeling, and `DBS`, which is based on the span frequency calculation in the Diverse Beam Search generation output. For white-box methods, LR refers to logistic regression, and GB refers to gradient boosting.

6 Results

6.1 Baselines

The results of baseline evaluation are shown in Table 1. The FAVA model performed with IoU score at the relatively low level of 26.49 on validation and 27.43 on test, which is significantly lower than the Black-box$_{\text{Base}}$ level with the same settings, without any soft labeling strategy. This shows that the FAVA taxonomy is probably not complete enough and does not observe many errors that LLMs generate. The FactScore$_{\text{GPT-4o}}$ shows relatively low performance, while the trained ModernBERT achieves an IoU score of 32.87 on the validation set and 33.35 on the test set.

When considering LLM-based methods, the GPT-4o and Phi-4 models that were used in 3-shot mode with randomly sampled examples show the best results compared to all the baselines.

6.2 Black-box

First, we perform ablation study of several LLMs to select the best performing model for hallucination detection in the Black-box pipeline (Table 7 in Appendix F). In contrast to results obtained in 1-shot setting, Qwen2.5-7B-it outperforms all other considered LLMs, even 14B Phi-4 model.

Next, we performed a soft-labeling hyperparameter ablation study (Table 5 in Appendix C). We found that the best IoU is observed with sampling 5 hypotheses using Diverse Beam Search along with a diversity penalty in the 1.0 level. Considering `Temp`, the best IoU is observed with sampling 5 hypotheses and temperature in 0.5.

Finally, we run `base` and `DBS` soft-labeling methods along with the fine-tuned Qwen2.5-7B-it on both the validation and test parts. On both validation and test, `DBS` is a best-performing soft-labeling strategy. Futhermore, this approach substantially outperforms all other methods described.

6.3 White-box

Detailed results of the white-box experiments across various languages and different training strategies are provided in Table 8 in Appendix G. Both the model-specific and model-agnostic approaches demonstrate improvements over the baseline methods. Notably, the model-agnostic approach outperforms the model-specific approach by 3% of IoU and by 11% of Cor, considering only the languages that are present in both the validation and test sets.

Additionally, the results indicate that a combination of all UQ methods yields robust improvements compared to using any single UQ method. Furthermore, the results with the GB model are slightly better that with the LR model. The final results, utilizing the model-agnostic approach trained on all UQ methods, are presented in Table 1.

6.4 Merging

In our work, we explore various approaches to combine the outputs of white-box and black-box methods. Specifically, we utilize logistic regression on predicted spans, gradient boosting with black-box predictions as features, and a simple weighted average approach.

Ultimately, the weighted average method, where the contribution of each method is controlled via parameters α and β, proved to be the most stable and effective fusion strategy. Our experiments revealed that the optimal values of α and β vary significantly across languages, and selecting them individually for each language leads to better results. To select the best hyperparameters, we perform a grid search on the validation set, selecting α and β that maximize IoU. A detailed breakdown of the results, including per-language optimal values, is provided in Table 2 in Appendix A.

7 Error Analysis

We conducted a detailed error analysis on a subset of English test examples to pinpoint the most common failures of our span-detection algorithms. We found that the vast majority of errors remain factual misclassifications. Notably, both the white-box and black-box methods tend to identify spans that are slightly longer. Although they still correctly cover the spans, they occasionally truncate entities – e.g. predicting "Ewald Klein in the 1930" instead of the distinct facts "chemist", "Ewald", "1930s" – a behavior more pronounced in the white-box method, which tends to select larger, statement-level spans over fine-grained annotations.

On the quantitative side, we measured in-accuracy in 14 languages (Table 9 in Appendix H), it evaluates whether the predicted answer includes the ground truth (Moskvoretskii et al., 2025). In many cases, the black-box method fully subsumes the expert spans – yielding higher in-accuracy but at the cost of over-segmentation. To mitigate both over- and under-segmentation, we experimented with two post-processing heuristics: (1) forcing inclusion or exclusion of partially labeled tokens and (2) stripping leading/trailing whitespace and punctuation from predicted spans. Fully including or excluding partially matched tokens uniformly reduced average IoU by 1.3% and 1.0% respectively, with no language benefiting. In contrast, applying only the punctuation cleanup heuristic increased IoU by 0.3% overall and by 2.2% for English (moving our English results from 9th to 6th place). These results demonstrate that simple span-boundary corrections can yield meaningful gains without retraining the core model.

Conclusion

We have shown that the lightweight white-box approach produces much better results that complicated baseline methods, even without relying on external knowledge.

The quality of our synthetic data generation

pipeline and the effectiveness of white-box approach is demonstrated by the high scores achieved by the merged method: our approach was ranked 3rd for four languages and within the top ten for the rest of the languages.

Limitations

Although synthetic data contains answers from LLMs of different sizes and architectures, only GPT-4o was used as a question-answer pairs generator and as a main annotator of hallucinations. This means that the annotation is probably not as objective as it could be if we used several proprietary models or even a group of crowdsourcers.

Since the uncertainty scores are poorly calibrated, we use supervised models in our white-box approach to both calibrate and combine several UQ methods. Consequently, the performance of this approach depends on the quality and size of the data available for training.

References

Marah I Abdin, Jyoti Aneja, Harkirat S. Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *CoRR*, abs/2412.08905.

Amos Azaria and Tom M. Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 967–976. Association for Computational Linguistics.

Julia Belikova, Evegeniy Beliakin, and Vasily Konovalov. 2024. JellyBell at TextGraphs-17 shared task: Fusing large language models with external knowledge for enhanced question answering. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 154–160, Bangkok, Thailand. Association for Computational Linguistics.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2024. Do androids know they're only dreaming of electric sheep? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4401–4420, Bangkok, Thailand. Association for Computational Linguistics.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. Association for Computational Linguistics.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *CoRR*, abs/2207.05221.

Nikita Krayko, Ivan Sidorov, Fedor Laputin, Alexander Panchenko, Daria Galimzianova, and Vasily Konovalov. 2025. Rurage: Robust universal rag evaluator for fast and affordable qa performance testing. In *Advances in Information Retrieval*, pages 135–145, Cham. Springer Nature Switzerland.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Ivan Maksimov, Vasily Konovalov, and Andrei Glinskii. 2024. DeepPavlov at SemEval-2024 task 6: Detection of hallucinations and overgeneration mistakes with an ensemble of transformer-based models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 274–278, Mexico City, Mexico. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *CoRR*, abs/2401.06855.

Daniil Moskovskiy, Sergey Pletenev, and Alexander Panchenko. 2024. LLMs to replace crowdsourcing for parallel data creation? the case of text detoxification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14361–14373, Miami, Florida, USA. Association for Computational Linguistics.

Daniil Moskovskiy, Nikita Sushko, Sergey Pletenev, Elena Tutubalina, and Alexander Panchenko. 2025. SynthDetoxM: Modern LLMs are few-shot parallel detoxification data annotators. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5714–5733, Albuquerque, New Mexico. Association for Computational Linguistics.

Viktor Moskvoretskii, Maria Lysyuk, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. 2025. Adaptive retrieval without self-knowledge? bringing uncertainty back home. *Preprint*, arXiv:2501.12835.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Elisei Rykov, Yana Shishkina, Ksenia Petrushina, Ksenia Titova, Sergey Petrakov, and Alexander Panchenko. 2024. SmurfCat at SemEval-2024 task 6: Leveraging synthetic data for hallucination detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 869–880, Mexico City, Mexico. Association for Computational Linguistics.

Maksim Savkin, Timur Ionov, and Vasily Konovalov. 2025. SPY: Enhancing privacy with synthetic PII detection dataset. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 236–246, Albuquerque, USA. Association for Computational Linguistics.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Kirill Grishchenkov, Sergey Petrakov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2024. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *arXiv preprint arXiv:2406.15627*.

Roman Vashurin, Maiya Goloburda, Preslav Nakov, Artem Shelmanov, and Maxim Panov. 2025. Cocoa: A generalized approach to uncertainty quantification by integrating confidence and consistency of llm outputs. *arXiv preprint arXiv:2502.04964*.

Artem Vazhentsev, Ekaterina Fadeeva, Rui Xing, Alexander Panchenko, Preslav Nakov, Timothy Baldwin, Maxim Panov, and Artem Shelmanov. 2024. Unconditional truthfulness: Learning conditional dependency for uncertainty quantification of large language models. *arXiv preprint arXiv:2408.10692*.

Artem Vazhentsev, Lyudmila Rvanova, Ivan Lazichny, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2025. Token-level density-based uncertainty quantification methods for eliciting truthfulness of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2246–2262, Albuquerque, New Mexico. Association for Computational Linguistics.

Raúl Vázquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sánchez-Vega, Alessandro Raganato, Jindřich Libovický, Jussi Karlgren, Shaoxiong Ji, Jindřich Helcl, Liane Guillou, Ona de Gibert, Jaione Bengoetxea, Joseph Attieh, and Marianna Apidianaki. 2025. SemEval-2025 Task 3: Mu-

SHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

A Merging Experiments

The hyperparameters were selected using the following grid: α from 0.1 to 1.0 with a step of 0.015, $\beta = 1 - \alpha$, and threshold from 0.05 to 0.65 with a step of 0.03. For each language, we selected the hyperparameters that maximized the IoU on the validation set. If the objective was to maximize the product of IoU and Cor on validation, the average IoU on the test set slightly decreased (to 57.66%), but Cor significantly improved (up to 58.31%). Additionally, the experiments revealed that each language requires its own set of hyperparameters; otherwise, if hyperparameters are tuned to maximize the average metrics across all languages, the merging results are worse than those of a single black-box method.

Language	α	β	**threshold**	**val**		**test**	
				IoU	Cor	IoU	Cor
ar	0.31	0.69	0.43	65.39	67.72	60.57	57.08
ca	0.47	0.53	0.46	-	-	67.27	57.40
cs	0.62	0.38	0.27	-	-	47.50	45.77
de	0.34	0.66	0.37	57.32	53.30	57.01	58.26
en	0.50	0.50	0.33	49.95	53.94	50.32	59.34
es	0.63	0.37	0.56	50.49	38.18	43.16	55.20
eu	0.62	0.38	0.33	-	-	51.96	45.19
fa	0.69	0.31	0.59	-	-	64.17	46.36
fi	0.13	0.87	0.30	58.18	53.57	62.92	55.38
fr	0.41	0.59	0.30	50.24	48.25	55.23	54.94
hi	0.77	0.23	0.49	67.86	58.94	71.19	60.79
it	0.15	0.85	0.37	66.37	52.96	70.38	61.99
sv	0.22	0.78	0.33	60.06	44.16	62.02	46.85
zh	0.13	0.87	0.24	48.14	35.45	48.97	35.70
Mean	-	-	-	57.40	50.65	58.05	52.88

Table 2: Results of hyperparameter tuning for the weighted average of white-box and black-box results.

B Black-box Training Details

The black-box model is trained on a dataset mixture created by combining the synthetic data, FAVA and RAGTruth datasets. Details about each dataset are shown in Table 3.

Dataset	**# of training samples**
Synthetic data	52 271
FAVA	27 364
RAGTruth	4699
Total	84 334

Table 3: Training dataset mixture details.

LLMs training for the black-box approach was performed on a single 8xA100 node with DeepSpeed Stage 2 optimization. The global batch size was 32 samples. All models were trained in a fixed setup with 4 epochs and a linear learning rate scheduler. All details on custom hyperparameters are shown in Table 4.

Hyperparameter	**Value**
learning_rate	1e-5
num_train_epochs	4
lr_scheduler_type	linear
warmup_ratio	0.3
gradient_accumulation_steps	32
batch_size	1
deepspeed stage	2

Table 4: Black-box training hyperparameters, remaining hyperparameters follow HuggingFace Trainer defaults.

C Soft Labeling Details

The soft labeling methods ablation for the black-box approach on the validation subset is shown in Table 5.

The `DBS` approach strongly outperforms the baseline, `Logit`, and `Temp` methods. Sampling more hypotheses, 10 instead of 5, only slightly improves IoU when the diversity penalty is 0.5. However, this effect is not relevant when the diversity penalty is greater than 0.5. We found that the best IoU is observed with 5 hypotheses and $\alpha = 1.0$.

The `Temp` approach shows quite robust results. Scaling temperature and sampling more hypotheses negatively affects the results for this soft labeling approach. The best IoU for `Temp` is 49.19 with 5 hypotheses and a temperature level of 0.5. Baselines

Soft Labeling Method	**val**	
	IoU	Cor
Base	46.78	43.58
Logit	47.00	43.38
DBS $n = 5, \alpha = 0.5$	52.52	49.54
DBS $n = 5, \alpha = 0.75$	53.39	**50.74**
DBS $n = 5, \alpha = 1.0$	**53.43**	49.95
DBS $n = 10, \alpha = 0.5$	53.30	50.59
DBS $n = 10, \alpha = 0.75$	52.84	47.59
DBS $n = 10, \alpha = 1.0$	51.53	45.39
Temp $n = 5, t = 0.5$	49.19	47.01
Temp $n = 5, t = 1.0$	49.18	46.91
Temp $n = 5, t = 1.5$	49.18	47.11
Temp $n = 10, t = 0.5$	48.31	47.62
Temp $n = 10, t = 1.0$	48.14	47.61
Temp $n = 10, t = 1.5$	48.07	47.62

Table 5: Ablation on soft labeling methods for Black-box. n stands for `num_beams`, α for `diversity_penalty`, t for `temperature`.

D Black-box Pipeline Architecture

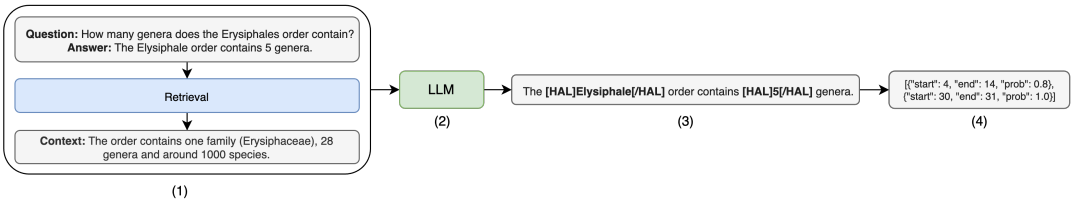

Figure 2: Black-box pipeline architecture. (1) Using the input question, we retrieve additional context and (2) pass it along with the question and answer to the fine-tuned LLM. (3) The model then detects and highlights hallucinations. (4) Finally, we post-process the LLM outputs and perform soft labeling to assign span probabilities.

E ModernBERT Training Details

As the black-box model, ModernBERT was trained on the full dataset mixture described in Appendix B. Training was performed on a single A100 40GB GPU. Details on the training hyperparameters are given in Table 6.

Hyperparameter	Value
learning_rate	1e-5
num_train_epochs	15
weight_decay	0.01
gradient_accumulation_steps	40
batch_size	1

Table 6: Encoder training hyperparameters.

F LLM Ablation

We used LLMs of different sizes and architectures for the black-box pipeline (Abdin et al., 2024; Yang et al., 2024). The results of our comparison are shown in Table 7. We used the Mu-SHROOM validation subset for ablation. We did not ablate different soft-labeling methods here as the goal of this ablation is to select the best base LLM for other ablations. Thus, according to our `Base` soft labeling approach, the probability of each span is assigned as 1.0. For each LLM, the training hyperparameters and the data set mixture were the same as described in the Appendix B.

We found that while Phi-4 and Qwen2.5-14B-Instruct are the largest models in our study, they are not the best performing LLMs in this setting. By averaging the IoU and Cor scores, Qwen2.5-7B-Instruct LLM outperforms all other models.

Although Qwen2.5-3B-it is the smallest model of the observed ones, it is only slightly inferior to larger models for French and Italian. For English and Swedish it even surpasses all observed checkpoints. Overall, its IoU score at the level of 37.89 is close to the IoU score of the best performing Qwen2.5-7B-it.

Also, the 14B model performs slightly better in Spanish, German, French, and Swedish, but yields significantly to the 7B model in Hindi and Finnish.

Phi-4 shows comparable performance to the 7B and 14B models for German and Swedish. For all other languages, including English, Phi-4 shows inferior performance.

LLM	ar		es		fr		de		it		hi		zh		en		fi		sv		Mean	
	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor
Phi-4	44.29	13.36	36.07	18.84	29.18	15.97	43.62	19.81	34.58	10.43	9.25	6.09	13.74	-0.58	30.13	13.37	34.02	11.12	44.25	3.83	31.91	11.22
Qwen2.5-3B-it	50.48	53.85	42.69	33.5	36.58	38.18	41.63	45.78	48.95	49.09	12.92	11.77	19.36	10.41	**38.61**	31.49	38.96	37.03	**48.74**	24.25	37.89	33.54
Qwen2.5-7B-it	55.68	54.97	**46.5**	35.24	37.38	**41.22**	46.09	44.72	**49.29**	**50.14**	**18.63**	**15.02**	20.64	13.84	36.35	**34.27**	**45.11**	**42.76**	45.01	28.22	**40.06**	36.04
Qwen2.5-14B-it	**61.10**	**58.81**	43.43	**38.46**	**38.91**	40.68	**46.60**	**46.32**	47.11	48.91	13.58	12.56	**26.71**	**18.90**	34.60	33.58	38.53	37.62	45.15	**33.79**	39.57	**36.96**

Table 7: LLM ablation in Black-box pipeline. We fine-tuned these LLMs on our dataset mixture including synthetic data and evaluated using Mu-SHROOM validation subset.

G White-box Results

The detailed results with the white-box methods are presented in Table 8. These results indicate that model-agnostic training of the GB model, leveraging all UQ methods, achieves the best average results on both validation and test datasets.

Additionally, we explore the use of the features extracted from the attention matrices, as proposed by Vazhentsev et al. (2024). The LR model trained on these features demonstrate the best performance on several language, such as English and Finnish. However, it is important to note that the number of extracted features varies across models due to differences in the number of layers and attention heads. As a result, experiments with this approach are conducted using a model-specific strategy, considering only the languages presented in the validation set.

Method	Scaling method	ar		es		fr		de		it		hi		zh		en		fi		cs		ca		fa		eu		sv		Mean	
		IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor
val																															
MTP, TE, CCP	LR, specific	42.92	37.88	**32.85**	34.91	37.99	25.11	44.42	35.63	52.14	41.56	47.13	42.02	47.08	17.25	32.78	31.44	48.45	35.85	-	-	-	-	-	-	-	-	58.01	25.97	44.38	32.76
MTP, TE, CCP	LR, agnostic	42.37	49.61	27.65	36.15	42.01	31.62	50.33	40.4	56.43	45.21	44.07	43.13	47.33	23.06	29.94	38.83	51.25	42.23	-	-	-	-	-	-	-	-	**59.64**	43.92	45.10	39.42
MTP, TE, CCP	GB, specific	46.62	50.80	29.25	**38.26**	41.91	30.99	47.61	40.08	54.29	42.47	53.31	42.51	47.65	25.59	35.65	31.49	46.79	38.44	-	-	-	-	-	-	-	-	57.23	25.99	46.03	36.66
MTP, TE, CCP	GB, agnostic	48.54	**63.48**	30.17	35.87	**43.15**	**37.95**	**52.35**	**46.17**	**60.28**	**49.19**	**53.72**	**48.93**	47.75	**31.43**	33.88	41.19	55.07	**47.23**	-	-	-	-	-	-	-	-	58.01	**48.09**	**48.29**	**44.95**
CCP	GB, agnostic	**54.34**	62.19	31.94	33.01	39.87	32.95	45.82	38.11	56.56	46.41	51.87	44.59	47.13	31.41	32.04	39.19	51.16	41.22	-	-	-	-	-	-	-	-	53.15	37.55	46.39	40.66
MTP	GB, agnostic	42.04	45.40	29.61	35.81	42.27	33.01	45.43	38.98	57.79	44.96	45.03	40.79	46.94	21.27	32.09	35.15	52.11	40.69	-	-	-	-	-	-	-	-	56.55	35.29	44.99	37.13
TE	GB, agnostic	49.44	54.41	28.34	34.40	41.43	34.40	50.32	41.20	57.21	47.48	46.00	43.54	47.58	20.21	34.92	35.26	52.30	47.03	-	-	-	-	-	-	-	-	56.59	44.19	46.41	40.21
Att. features	LR, specific	51.86	48.74	25.74	33.88	40.21	28.77	45.57	42.57	50.55	41.59	50.50	46.47	**51.91**	30.08	**56.75**	**42.45**	**56.35**	42.60	-	-	-	-	-	-	-	-	-	-	47.71	39.68
test																															
MTP, TE, CCP	LR, specific	56.26	**57.51**	19.64	37.51	50.64	43.70	37.94	30.78	52.61	52.45	36.31	33.20	48.05	21.13	38.51	37.14	54.71	43.04	-	-	-	-	-	-	-	-	53.63	38.05	44.83	39.45
MTP, TE, CCP	LR, agnostic	55.78	57.38	20.98	**42.07**	55.79	47.94	40.58	39.93	52.81	55.29	42.86	44.33	49.01	25.29	38.42	42.48	58.45	48.21	34.05	32.15	35.79	44.17	24.05	18.59	38.72	32.14	51.91	41.03	42.80	40.79
MTP, TE, CCP	GB, specific	**60.56**	57.29	20.61	38.45	53.74	47.94	39.89	36.84	55.05	53.42	42.28	40.09	48.80	29.47	33.41	42.44	55.39	42.47	-	-	-	-	-	-	-	-	**56.39**	38.52	46.61	42.69
MTP, TE, CCP	GB, agnostic	60.01	56.72	24.79	43.83	**58.61**	**51.76**	44.68	43.03	**58.97**	**57.77**	**49.28**	**46.28**	**49.28**	29.60	39.98	45.72	59.31	50.19	**37.84**	**33.46**	38.77	**48.03**	**27.47**	30.45	36.51	**33.21**	54.19	**41.13**	**45.69**	**43.67**
CCP	GB, agnostic	60.16	57.15	21.37	37.07	55.45	48.04	39.77	39.96	49.87	53.06	37.73	42.73	48.80	31.52	37.44	42.01	55.92	43.29	37.56	26.39	**39.31**	45.27	25.84	**32.71**	**41.15**	31.39	54.87	34.75	43.23	40.38
MTP	GB, agnostic	59.92	56.68	16.55	39.77	57.42	49.69	42.02	37.17	53.95	54.93	42.93	43.00	48.37	19.71	38.19	42.72	58.35	46.60	34.56	32.23	35.54	43.81	22.78	16.47	39.74	32.58	52.64	37.35	43.07	39.48
TE	GB, agnostic	60.06	56.84	16.39	41.70	58.37	50.85	34.90	36.08	53.48	56.20	41.12	44.40	49.03	19.42	41.85	43.55	58.48	48.01	33.63	31.64	35.74	43.83	22.11	20.36	35.04	32.51	55.42	38.69	42.55	40.29
Att. features	LR, specific	42.85	45.76	**26.94**	41.71	47.46	39.51	**49.45**	**49.22**	53.54	52.82	44.04	41.34	46.47	**34.03**	**45.95**	**49.82**	**60.30**	**56.07**	-	-	-	-	-	-	-	-	-	-	46.33	45.59

Table 8: Experimental results with various white-box methods. LR refers to logistic regression, and GB refers to gradient boosting. For the model-specific methods, part of the test set results is missing because these languages and models were not present in the validation set. The average test set results for the model-specific methods are based on the less number of languages, as four languages (cs, ca, fa, eu) are missed in the validation set. The table also includes ablation experiments where only one of the uncertainty quantification methods was retained. The best performing method is in bold, the second-best is underlined.

H Span-detection Error Analysis

Lang	White-box	Black-box	Merged
FR	60.00	62.67	60.00
ES	32.24	55.26	49.34
HI	41.33	77.33	72.00
AR	54.00	55.33	55.33
CA	51.00	73.00	73.00
ZH	57.33	49.33	74.67
IT	48.00	80.00	82.67
DE	30.67	57.33	66.00
FA	52.00	62.00	58.00
SV	42.18	72.11	74.83
EU	27.27	68.69	66.67
EN	50.00	60.39	51.95
CS	54.00	49.00	54.00
FI	33.33	72.00	80.00

Table 9: Results of the in-accuracy metric across different languages for the proposed span detection methods.